UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/817,275	04/01/2004	Michael A. Howitz	16010-07728	2281

758 7590 05/16/2007

FENWICK & WEST LLP
SILICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW, CA 94041

EXAMINER	
SAN JUAN, MARTINJERIKO P	
ART UNIT	**PAPER NUMBER**
2109	

MAIL DATE	DELIVERY MODE
05/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/817,275	HOWITZ ET AL.	
	Examiner	Art Unit	
	Martin Jeriko P. San Juan	2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 April 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This is response to the following case application:

Non-provisional Application No. 10/817275 filed on April 1, 2004.

Specification

1. The disclosure is objected to because of the following informalities:

   a. Page 6, Ln 1 -- "...confirmation of the storage of the [date] will be...." should be read as "...confirmation of the storage of the [data] will be..."

   b. Page 13, Par 0033, Ln 6 – the "single sing-on server 13-" should be "single sing-on server 130."

   Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

1. Claim 1-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Roberts et al. [US PN 6295551 B1].

   a. Based on independent claim 1, Roberts et al. teach a cross-platform single sign-on system for sharing user data across computers on a plurality of

computing platforms, the system comprising: an authentication module for authenticating a user at the beginning of a computing session [Col 11, Ln 26-31]; an interface module configured to receive requests for authentication and non-authentication data associated with the user from a plurality of applications on the plurality of computing platforms [Col 10, Ln 52-67] [*The applets are the interface modules between computers that can run a plurality of objects through the applet (plurality of applications) for sharing content. Col 10, Ln 52-55. The server ascertaining what type of computer is requesting a download of one of applets inherently teaches an interfacing module receiving requests for a second computer to communicate thereby involving the transfer of authentication and non-authentication data. Since there are many type of interfacing applets, which can operate from different platforms, the passage teaches a computer requesting connection* coming from a plurality of applications on the plurality of computing platforms.] and, based upon authentication of the user at the beginning of the computing session and responsive to the requests, to automatically provide authentication and non-authentication data to the plurality of applications throughout the computing session [Col 11, Ln 5-15] [Col 8, Ln 18-20] [*A connection session will be established only when the validation/authentication is successful. Upon success, it is inherent that authentication and non-authentication data will be transferred automatically to enable shared content through the applet running a plurality of objects/applications.*]; and a data registry in communication with the interface module for storing and providing

authentication data and non-authentication data responsive to requests made by the plurality of applications [Fig 1, Itm 20] [*The server will inherently have a data registry to keep track of different computers requesting different types of sessions/connections. Through the applet, the server is inherently storing and providing authentication and non-authentication data to establish and maintain a connection/session with another computer.*]

b. With regard to dependent claim 2, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein web services technologies are used to transmit requests for authentication and non-authentication data from a plurality of computer systems hosting the plurality of applications to the interface module [Col 8, Ln 52-65] [*The different objects are the different applications available in a connection/session.*].

c. With regard to dependent claim 3, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein the non-authentication data includes state information reflecting a state of a selected application on a first computer accessed by the user that can be retrieved when the selected application is being accessed from a second computer [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38] [*For different application objects showing identical/synchronous information, it is inherent that state information is being transferred.*].

d. With regard to dependent claim 4, Roberts et al. teach the cross-platform single sign-on system of claim 3, wherein the selected application is being accessed from a second computer by a second user [Col 13, Ln 61].

e. With regard to dependent claim 5, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein the interface module is further configured to receive requests to store authentication and non-authentication data associated with the user from a plurality of applications on a plurality of computing platforms in the computing system and, based upon authentication of a user at the beginning of a computing session and responsive to the requests, to store the data to the data registry [*Authentication data is being supplied by the user to a server to be stored because session information is being tracked.* Col 14, Ln 48-59] [*Non-authentication data, eg. state information, has been stored by the server if the server needs to wait for an acknowledgement for the shared content to be updated.* Col 8, Ln 25-26].

f. With regard to dependent claim 6, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein the interface module formats data queries to the data registry in accordance with a data exchange protocol accepted by the data registry [*Protocol is Javascript communication.* Col 12, Ln 60].

g. With regard to dependent claim 7, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein the data registry is further configured to receive requests for authentication and non-authentication data directly from the

plurality of applications on the plurality of computing platforms, and for the requested data to be retrieved from the data registry responsive to the requests [Col 12, Ln 39-50] [Col 14, Ln 48-59] [*The data registry is hosted in the server and thus receive requests directly from other computers requesting a connection/session. The passage teaches the many different kinds of authentication and non-authentication data being transferred.*]

h. With regard to dependent claim 8, Roberts et al. teach the cross-platform single sign-on system of claim 1, wherein a request to retrieve authentication and non-authentication data associated with the user is sent responsive to an event trigger activated during the user's computing session [Col 8, Ln 45-51].

i. With regard to dependent claim 9, Roberts et al. teach the cross-platform single sign-on system of claim 8, wherein the event trigger comprises at least one of: the authentication of a user, a user command, and the passage of a pre-determined interval of time [Col 8, Ln 1-17].

j. With regard to dependent claim 10, Roberts et al teach the cross-platform single sign-on system of claim 1, wherein the interface module and authentication module are commonly hosted on a single computer [Fig 1, Itm 12, Itm 24, Itm 20].

k. With regard to dependent claim 11, Roberts et al teach the cross-platform single sign-on system of claim 1, wherein at least one of the plurality of computing platforms differs from another at least another of the plurality of computing platforms [*An operating platform used by the first domain differs from*

an operating platform used by the second domain is taught since JAVA objects are being utilized enabling a "cross-platform" system. Col 8, Ln 52-65].

l. With regard to dependent claim 12, Roberts et al teach the cross-platform single sign-on system of claim 1, further comprising: a caching module for storing non-authentication data generated by an application in the local cache of the computer hosting the application when the computer is disconnected from the computing system; and a synchronizing module for sending non-authentication data stored in the local cache to the data registry when the computer is connected to the computing system [Col 9, Ln 25-38] [*The "persistent" applet is loaded into a local cache and being persistent inherently teaches a synchronizing module that will keep computers in-synchronization as long as the computer remains in a session. It inherently teaches protecting sessions from network interruptions since it explicitly states that it remains in the cache unless it gets removed by the user computer.*]

m. With regard to dependent claim 13, Roberts et al teach the cross-platform single sign-on system of claim 1, wherein the authentication module is configured to detect that a user is logging on to the system for the first time [Col 10, Ln 56-67] [*The server computer ascertaining whether a download of applet is needed inherently teaches a user logging on for the first time.*], further comprising: a verification module in communication with the data registry for verifying the identity of the user [Col 11, 26-31]; a password capture utility launched responsive to the successful verification of the user's identity for creating a global

user id and password for the user with which the user can be logged on to the cross-platform single sign-on system [Col 11, Ln 5-14] [*A validation of user identity using a password system inherently teaches a password capture utility.*], capturing user authentication information associated with applications launched during the user's computing session, and storing the authentication information in the data registry [Col 14, Ln 48-59] [*The session box inherently teaches that user authentication information is captured and stored in data registry/database.*]

n. With regard to dependent claim 14, Roberts et al teach a data registry for storing and providing data across a computing system, the data registry comprising [*There is implementation of database/data registry.* Col 11, Ln 27]: a plurality of user data entries [Col 11, Ln 25-31], each of the user data entries describing a unique user of an computing system comprised of a plurality of computing platforms and a plurality of applications [*There exist different types of users with different kinds of applications, and types of computer, ie. sales representative, customer, administrator.* (Col 11, Ln 5) (Col 10, Ln 57)]; a plurality of authentication entries associated with each of the user data entries for authenticating the user on the plurality of applications of the computing system [Col 11, Ln 5-14] [*This passage inherently teach a plurality of authentication entries because the server can ascertain and validate which type of user is making a request.*]; and a plurality of non-authentication attributes and attribute entries associated with each of the user data entries in which information about a user's use of an application can be preserved [Col 10, Ln 56-67] [Col 14, Ln 48-

59] [*Passage inherently teach a plurality of non-authentication attributes and attribute entries because the server can ascertain what applet and associated objects are needed by a user making the request. Also the passage teaches a session box which inherently capture information about a user's use of an application.*].

o. With regard to dependent claim 15, Roberts et al. teach the data registry of claim 14, wherein the non-authentication data includes state information for one of the plurality of applications, whereby a user may switch between a first computer and a second computer and preserve the state of a selected application accessed using the first computer when accessing the selected application from the second computer [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38].

p. With regard to dependent claim 16, Roberts et al. teach the data registry of claim 14, wherein the non-authentication data includes configuration information for one of the plurality of applications with which a user's application environment can be customized [Col 12, Ln 66 – Col 13, Ln 60].

q. With regard to dependent claim 17, Roberts et al. teach the data registry of claim 14, further comprising an interface module that receives web service requests for storing and providing data from one of the plurality of applications and, responsive to the requests, saves the data to the data registry [Col 12, Ln 39-50] [*The data registry is in the server that also runs the applet. This passage cites command scripts which teaches web service requests.*]

r. Based on independent claim 18, Roberts et al. teaches a method of sharing data across a computing system, the method comprising: subsequent to an initial authentication of a user [Col 11, Ln 26-31], receiving requests to authenticate the authenticated user from a plurality of applications on a plurality of computing platforms being accessed by the authenticated user [Col 11, Ln 14-15] [*This passage teaches that "others" understood to have been initially authenticated can join an on-going session. This inherently teach receiving requests to authenticate the authenticated user.*]; automatically authenticating the authenticated user to the plurality of applications being accessed by the authenticated user responsive to the initial authentication of the user [Col 11, Ln 31-35] [*"Automatically authenticating the authenticated user responsive to the initial authentication of the user" is interpreted as automatically authenticating a second or subsequent requests by responding/undergoing the same process as usually done for the initial authentication. This passage teaches that the others who had been initially authenticated undergoes another authentication for subsequent requests.*]; receiving non-authentication data provided by a first instance of the authenticated user using an application in a first domain [Col 8, Ln 18-20]; storing the non-authentication data provided by the first instance of the authenticated user using the application in the first domain [*It is inherent that non-authentication data has been stored if the server needs* to wait for an acknowledgement for the shared content to be updated. Col 8, Ln 25-26]; receiving a request for non-authentication data from a second instance of the

application in a second domain [Col 10, Ln 56-67] [*This is equivalent to requesting a new or ongoing connection/session.*]; and supplying the requested non-authentication data provided by the first instance of the application in the first domain to the second instance of the application in the second domain [Col 8, Ln 25-26].

s. With regard to dependent claim 19, Roberts et al. teaches the method of claim 18, wherein the second instance of the application in the second domain is associated with a second user [Col 13, Ln 61].

t. With regard to dependent claim 20, Roberts et al. teaches the method of claim 18, further comprising: receiving log on information from a user [Col 10, Ln 56-67]; determining that the user is logging on to the computing system for the first time [*The step of determining that the user is logging on to the computing system for the first time is being taught since session information is being "tracked" including any associated passwords (Col 14, Ln 48-59) and that the call request can be routed by the server into queues based upon attributes of the user (Col 15, Ln 7-8).*]; subsequent to the determination that a user is logging on to the computing system for the first time, verifying the identity of the user [Col 20, Ln 37-51]; prompting the user to supply a user id and password [Col 20, Ln 41]; providing the user id and password supplied by the user to a data registry to be stored therein [*Authentication data is being supplied by the user to a data registry in the server to be stored is taught because session information is being "tracked." Col 14, Ln 48-59*]; capturing application authentication information

provided by the user during the computing session; storing the application authentication information provided by the user during the computing session in the data registry wherein the data registry is configured to store authentication and non-authentication data [Col 14, Ln 48-59].

u. With regard to dependent claim 21, Roberts et al. teaches the method of claim 18, wherein an operating platform used by the first domain differs from an operating platform used by the second domain [*Operating platform used by the first domain differs from an operating platform used by the second domain is taught since JAVA objects are being utilized enabling a "cross-platform" system. Col 8, Ln 52-65*].

v. With regard to dependent claim 22, Roberts et al. teaches the method of claim 18, further comprising storing authentication data in the data registry [*There is a method step of storing authentication data in a data registry because when a requester logs-on to the server, the server validates the requester against the database. Col 11, Ln 26-31*].

w. With regard to dependent claim 23, Roberts et al. teaches the method of claim 18, wherein the non-authentication data provided by the first instance of the application in the first domain comprises configuration information for customizing a user's application environment [Col 12, Ln 66 – Col 13, Ln 60].

x. With regard to dependent claim 24, Roberts et al. teaches the method of claim 18, wherein the non-authentication data provided by the first instance of the application in the first domain includes state information with which the user's

application state from the first instance of the application in the first domain can be maintained to the second instance of the application in the second domain [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38].

y. With regard to dependent claim 25, Roberts et al. teaches the method of claim 18, wherein storing the non-authentication data comprises: configuring a non-authentication data attribute; storing a value for the non-authentication data attribute associated with the user; and responsive to a request identifying the non-authentication data attribute, providing the value of the non-authentication data attribute to a requesting application [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38].

z. With regard to dependent claim 26, Roberts et al. teaches the method of claim 18, wherein the request for non-authentication data associated with the authenticated user is generated responsive to a call trigger [Col 8, Ln 45-51].

aa. With regard to dependent claim 27, Roberts et al. teaches the method of claim 18, wherein the step of receiving non-authentication data provided by a first instance of an application used by the authenticated user comprises receiving the non-authentication data from a synchronizing module on a computer for sending non-authentication data from the local cache of the computer, the data having been stored in the local cache when the authenticated user was disconnected from the networked system [Col 9, Ln 25-38].

bb. With regard to dependent claim 28, The system of claim 1, wherein the non-authentication data comprises one of: configurations data, settings data, or

applications data, environment data [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38].

cc. With regard to dependent claim 29, The system of claim 1, wherein the non-authentication data comprises one of: a size of a window, the configuration of a tool bar, and the selection of open files [Col 12, Ln 66 – Col 13, Ln 60] [Col 14, Ln 48-59] [Col 18, Ln 35-38].

2. Claim 1-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Hickman [US PN. 6173332 B1].

a. Based on independent claim 1, Hickman teaches a cross-platform single sign-on system for sharing user data across computers on a plurality of computing platforms, the system comprising: an authentication module for authenticating a user at the beginning of a computing session [Fig 5, Itm 286] [Fig 6, Itm 316]; an interface module configured to receive requests for authentication and non-authentication data associated with the user from a plurality of applications on the plurality of computing platforms [Col 10, Ln 1-18] [*The virtual machine application can be launched in several ways from remote terminal computers, thereby inherently teaching an interface module.*] and, based upon authentication of the user at the beginning of the computing session and responsive to the requests, to automatically provide authentication and non-authentication data to the plurality of applications throughout the computing session [Col 21, Ln 27-30. *This passage explains the success of authentication.*]

[Col 19, Ln 9-16] [*Upon success, a client computer can have access and control to a NAC whose user's authentication and non-authentication data are transferred to the NAC to enable access and control.*]; and a data registry in communication with the interface module for storing and providing authentication data and non-authentication data responsive to requests made by the plurality of applications [Col 21, Ln 1-30] [*The CAC handles all requests made for accessing/controlling other NAC computers and therefore inherently teaches a data registry.*].

b. With regard to dependent claim 2, Hickman teaches the cross-platform single sign-on system of claim 1, wherein web services technologies are used to transmit requests for authentication and non-authentication data from a plurality of computer systems hosting the plurality of applications to the interface module [Col 7, Ln 52-58] [*TCP/IP are among web services technologies.*].

c. With regard to dependent claim 3, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the non-authentication data includes state information reflecting a state of a selected application on a first computer accessed by the user that can be retrieved when the selected application is being accessed from a second computer [Col 19, Ln 9-16].

d. With regard to dependent claim 4, Hickman teaches the cross-platform single sign-on system of claim 3, wherein the selected application is being accessed from a second computer by a second user [Col 19, Ln 4-16].

e. With regard to dependent claim 5, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the interface module is further configured to receive requests to store authentication and non-authentication data associated with the user from a plurality of applications on a plurality of computing platforms in the computing system and, based upon authentication of a user at the beginning of a computing session and responsive to the requests, to store the data to the data registry [Col 21, Ln 1-67] [*The CAC is the interface for hosting the necessary modules for receiving requests to connect to other NACs, and based upon success of authentications, enable access and control of NAC from a remote terminal computer. The CAC also inherently teaches the data registry.*]

f. With regard to dependent claim 6, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the interface module formats data queries to the data registry in accordance with a data exchange protocol accepted by the data registry [TCP/IP data packet transmission protocols, Col 7, Ln 51].

g. With regard to dependent claim 7, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the data registry is further configured to receive requests for authentication and non-authentication data directly from the plurality of applications on the plurality of computing platforms, and for the requested data to be retrieved from the data registry responsive to the requests [Col 21, Ln 1-67] [*The CAC inherently host the data registry. The CAC also*

manages connection/session between the NAC and remote computers. The CAC teaches receiving requests from other computers which is from a plurality of application on the plurality of computing platforms.].

h. With regard to dependent claim 8, Hickman teaches the cross-platform single sign-on system of claim 1, wherein a request to retrieve authentication and non-authentication data associated with the user is sent responsive to an event trigger activated during the user's computing session [Col 13, Ln 16-18].

i. With regard to dependent claim 9, Hickman teaches the cross-platform single sign-on system of claim 8, wherein the event trigger comprises at least one of: the authentication of a user, a user command, and the passage of a pre-determined interval of time [Col 13, Ln 16-18] [Col 13, Ln 44].

j. With regard to dependent claim 10, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the interface module and authentication module are commonly hosted on a single computer [Fig 1].

k. With regard to dependent claim 11, Hickman teaches the cross-platform single sign-on system of claim 1, wherein at least one of the plurality of computing platforms differs from another at least another of the plurality of computing platforms [Col 9, Ln 1-3].

l. With regard to dependent claim 12, Hickman teaches the cross-platform single sign-on system of claim 1, further comprising: a caching module for storing non-authentication data generated by an application in the local cache of the computer hosting the application when the computer is disconnected from the

computing system; and a synchronizing module for sending non-authentication data stored in the local cache to the data registry when the computer is connected to the computing system [Col 18, Ln 34] [Col 18, Ln 59-63] [*A caching module and a synchronizing module are inherently taught because this passage teaches redundancy and change updates. It teaches various contingency operations caused by various network interruption problems eg. crashes, power losses.*].

m. With regard to dependent claim 13, Hickman teaches the cross-platform single sign-on system of claim 1, wherein the authentication module is configured to detect that a user is logging on to the system for the first time [Fig 5, Itm 284], further comprising: a verification module in communication with the data registry for verifying the identity of the user [Col 21, Ln 10-15]; a password capture utility launched responsive to the successful verification of the user's identity for creating a global user id and password for the user with which the user can be logged on to the cross-platform single sign-on system [Col 21, Ln 27-31], capturing user authentication information associated with applications launched during the user's computing session, and storing the authentication information in the data registry [Col 21, Ln 27-31] [*Since the CAC has all the information, CAC inherently teaches such authentication information being stored in a data registry.*]

n. With regard to dependent claim 14, Hickman teaches a data registry for storing and providing data across a computing system, the data registry

comprising: a plurality of user data entries, each of the user data entries describing a unique user of an computing system comprised of a plurality of computing platforms and a plurality of applications [Col 21, Ln 27-31]; a plurality of authentication entries associated with each of the user data entries for authenticating the user on the plurality of applications of the computing system [Col 21, Ln 40] [Col 21, Ln 64-67]; and a plurality of non-authentication attributes and attribute entries associated with each of the user data entries in which information about a user's use of an application can be preserved [Col 22, Ln 10] [Col 22, Ln 20-24] [Col 22, Ln 29-31].

o. With regard to dependent claim 15, Hickman teaches the data registry of claim 14, wherein the non-authentication data includes state information for one of the plurality of applications, whereby a user may switch between a first computer and a second computer and preserve the state of a selected application accessed using the first computer when accessing the selected application from the second computer [Col 12, Ln 15-24] [*The CAC inherently teaches the data registry. This passage teaches the types of computer access which state information can be preserved when accessed from another remote computer.*].

p. With regard to dependent claim 16, Hickman teaches the data registry of claim 14, wherein the non-authentication data includes configuration information for one of the plurality of applications with which a user's application environment can be customized [Col 19, Ln 1-9].

q. With regard to dependent claim 17, Hickman teaches the data registry of claim 14, further comprising an interface module that receives web service requests for storing and providing data from one of the plurality of applications and, responsive to the requests, saves the data to the data registry [Col 21, Ln 1-9] [*The CAC being the interface receives request and manages NAC connection/session thereby teaching the storage and provision of data necessary for NAC access/control.*].

r. Based on independent claim 18, Hickman teaches a method of sharing data across a computing system, the method comprising: subsequent to an initial authentication of a user, receiving requests to authenticate the authenticated user from a plurality of applications on a plurality of computing platforms being accessed by the authenticated user [Col 13, Ln 41-42] [Col 14, Ln 26-30]; automatically authenticating the authenticated user [*Cryptographic keys are used established in the initial authentication for authenticating subsequent requests.*] to the plurality of applications being accessed by the authenticated user responsive to the initial authentication of the user [Col 13, Ln 41-42] [Col 14, Ln 26-30] [*This passage teaches both receiving request and automatically authenticating upon success.*]; receiving non-authentication data provided by a first instance of the authenticated user
using an application in a first domain [Col 19, Ln 1-9]; storing the non-authentication data provided by the first instance of the authenticated user using the application in the first domain [Col 19, Ln 24]; receiving a request for non-

authentication data from a second instance of the application in a second domain [Col 19, Ln 9-16]; and supplying the requested non-authentication data provided by the first instance of the application in the first domain to the second instance of the application in the second domain [Col 19, Ln 9-16]. [*When a NAC is access from remote computer, and when connection is established inherently teaches the receiving of data provided by a first instance, storing of the non-authentication data, receiving a request for data from a second instance, and then supplying the data.*]

s. With regard to dependent claim 19, Hickman teaches the method of claim 18, wherein the second instance of the application in the second domain is associated with a second user [Col 19, Ln 4-16].

t. With regard to dependent claim 20, Hickman teaches the method of claim 18, further comprising: receiving log on information from a user [Log-on information is part of configuration. (Fig 5, Step 286), (Fig 6)]; determining that the user is logging on to the computing system for the first time [Fig 5, Step 284] [*When determining configuration information is needed or not by a requesting user inherently teaches whether a user is logging for the first time.*]; subsequent to the determination that a user is logging on to the computing system for the first time, verifying the identity of the user [Col 21, Ln 10-30]; prompting the user to supply a user id and password [Col 21, Ln 10-30] [Col 21, Ln 40]; providing the user id and password supplied by the user to a data registry to be stored therein [Col 21, Ln 10-30]; capturing application authentication information provided by

the user during the computing session [Col 21, Ln 27-30]; storing the application authentication information provided by the user during the computing session in the data registry wherein the data registry is configured to store authentication and non-authentication data [Col 21, Ln 27-30].

u. With regard to dependent claim 21, Hickman teaches the method of claim 18, wherein an operating platform used by the first domain differs from an operating platform used by the second domain [Col 9, Ln 1-3].

v. With regard to dependent claim 22, Hickman teaches the method of claim 18, further comprising storing authentication data in the data registry [Col 13, Ln 40-44].

w. With regard to dependent claim 23, Hickman teaches the method of claim 18, wherein the non-authentication data provided by the first instance of the application in the first domain comprises configuration information for customizing a user's application environment [Col 19, Ln 1-23].

x. With regard to dependent claim 24, Hickman teaches the method of claim 18, wherein the non-authentication data provided by the first instance of the application in the first domain includes state information with which the user's application state from the first instance of the application in the first domain can be maintained to the second instance of the application in the second domain [Col 19, Ln 1-23].

y. With regard to dependent claim 25, Hickman teaches the method of claim 18, wherein storing the non-authentication data comprises: configuring a non-

authentication data attribute; storing a value for the non-authentication data attribute associated with the user; and responsive to a request identifying the non-authentication data attribute, providing the value of the non-authentication data attribute to a requesting application. [Col 19, Ln 11-23]

z. With regard to dependent claim 26, Hickman teaches the method of claim 18, wherein the request for non-authentication data associated with the authenticated user is generated responsive to a call trigger [Col 13, Ln 16-18].

aa. With regard to dependent claim 27, Hickman teaches the method of claim 18, wherein the step of receiving non-authentication data provided by a first instance of an application used by the authenticated user comprises receiving the non-authentication data from a synchronizing module on a computer for sending non-authentication data from the local cache of the computer, the data having been stored in the local cache when the authenticated user was disconnected from the networked system [Col 18, Ln 59-63].

bb. With regard to dependent claim 28, Hickman teaches the system of claim 1, wherein the non-authentication data comprises one of: configurations data, settings data, or applications data, environment data [Col 19, Ln 11-23].

cc. With regard to dependent claim 29, Hickman teaches the system of claim 1, wherein the non-authentication data comprises one of: a size of a window, the configuration of a tool bar, and the selection of open files [Col 19, Ln 11-23].

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Martin Jeriko P. San Juan whose telephone number is 571-272-7875. The examiner can normally be reached on M-F 7:30a - 5:00p EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Del Sole can be reached on 571-272-1130. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MJSJ

JOSEPH DEL SOLE
SUPERVISORY PATENT EXAMINER
5/14/07